1. Difference between display:none and visibility:hidden

visibility:hidden

- hides the element

- but it still takes up space in the layout.

display:none

-removes the element from the document

- It does not take up any space.

1. Flexbox

-[Flexbox](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout) is a one-dimensional layout method for arranging items in rows or columns.

-Items flex (expand) to fill additional space or shrink to fit into smaller spaces.

1. Meta data purpose

-The <meta> tag defines metadata about an HTML document. Metadata is data (information) about data.

- <meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.

-Metadata will not be displayed on the page, but the machine understands it .

-Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

1. Viewport

-The browser's [viewport](https://developer.mozilla.org/en-US/docs/Glossary/Viewport) is the area of the window in which web content can be seen.

-The basic properties of the "viewport" <meta> tag include:

* width
* height
* initial-scale:Controls the zoom level when the page is first loaded. Minimum: 0.1. Maximum: 10. Default: 1.
* Minimum-scale: Controls how much zoom out is allowed on the page.
* Maximum-scale:Controls how much zoom in is allowed on the page.
* User-scalable:Controls whether zoom in and zoom out actions are allowed on the page. Valid values: 0, 1, yes, or no.

1. CSS3 features compared to previous version

|  |  |  |
| --- | --- | --- |
| S.No. | CSS | CSS3 |
| 1 | CSS is capable of positioning texts and objects. | On the other hand, CSS3 is capable of making the web page more attractive and takes less time to create. CSS3 is backward compatible with CSS. |
| 2 | Responsive designing is not supported in CSS | CSS3 is the latest version, hence it supports responsive design. |
| 3 | CSS cannot be split into modules. | Whereas CSS3 can be breakdown into modules. |
| 4 | Using CSS, we cannot build 3D animation and transformation. | But in CSS3 we can perform all kinds of animation and transformations as it supports animation and 3D transformations. |
| 5 | CSS is very slow as compared to CSS3 | Whereas CSS3 is faster than CSS. |
| 6 | In CSS we have set of standard colors and it uses basic color schemes only. | Whereas CSS3 has a good collection of HSL RGBA, HSLA, and gradient colors. |
| 7 | In CSS we can only use single text blocks. | But in CSS3 we can use multi-column text blocks |
| 8 | CSS does not support media queries. | But CSS3 supports media queries |
| 9 | CSS codes are not supported by all types of modern browsers. | Being the latest version, CSS3 codes are supported by all modern browsers. |
| 10 | In CSS, designers have to manually develop rounded gradients and corners. | But CSS3 provides advanced codes for setting rounded gradients and corners |
| 11 | There is no special effect like shadowing text, text animation, etc. in CSS. The animation was coded in jQuery and JavaScript. | CSS3 has many advance features like text shadows, visual effects, and a wide range of font styles and colors. |
| 12 | In CSS, the user can add background colors to list items and lists, set images for the list items, etc. | Whereas CSS3 list has a special *display* property defined in it. Even list items also have counter reset properties. |
| 13 | CSS was developed in 1996. | CSS3 is the latest version of CSS and was released in 2005. |
| 14 | CSS is memory intensive.(high) | CSS3 memory consumption is low as compared to CSS. |

1. HTML vs HTML5

|  |  |
| --- | --- |
| ****HTML**** | ****HTML5**** |
| HTML does not provide native audio and video support. | HTML5 provides native audio and video support. |
| HTML only supports vector graphics if used in conjunction with different technologies like**[Flash](https://www.computerhope.com/jargon/f/flash.htm" \t "https://www.hostinger.in/tutorials/_blank)**, **[VML](https://www.w3.org/TR/NOTE-VML" \t "https://www.hostinger.in/tutorials/_blank)**, or**[Silverlight](https://www.microsoft.com/silverlight/" \t "https://www.hostinger.in/tutorials/_blank)**. | HTML5 supports SVG (Scalable Vector Graphics), Canvas, and other virtual vector graphics. |
| HTML allows inline MathML and SVG in text with restricted use. | HTML5 allows inline **[MathML](https://www.w3.org/Math/whatIsMathML.html" \t "https://www.hostinger.in/tutorials/_blank)** and **[SVG](https://www.bitdegree.org/learn/html5-svg" \t "https://www.hostinger.in/tutorials/_blank)** in text |
| HTML doesn’t allow users to draw shapes such as circles, triangles, and rectangles. | HTML allows users to draw shapes such as circles, triangles, and rectangles. |
| HTML only uses browser cache and cookies to store data temporarily. | HTML5 uses web SQL databases, local storage, and application cache for storing data temporarily. |
| JavaScript and browser interface run in the same thread. | JavaScript and browser interface run in separate threads. |
| Longer **[document type declaration](https://www.w3schools.com/tags/tag_doctype.asp" \t "https://www.hostinger.in/tutorials/_blank)**. | Shorter document type declaration. |
| Longer character encoding declaration. Uses the ASCII **[character set](https://www.w3schools.com/charsets/" \t "https://www.hostinger.in/tutorials/_blank)**. | Shorter **[character encoding](https://www.w3.org/International/questions/qa-what-is-encoding" \t "https://www.hostinger.in/tutorials/_blank)** declaration. Uses the UTF-8 character set. |
| Compatible with almost all browsers. | Only compatible with newer browsers, considering there are many new tags and elements which only some browsers support. |
| Built based on **[Standard Generalized Markup Language](https://www.w3.org/TR/WD-html40-970708/intro/sgmltut.html" \t "https://www.hostinger.in/tutorials/_blank)** (SGML). | HTML5 has improved parsing rules providing enhanced compatibility. |

7)

|  |  |
| --- | --- |
| CSS Transitions | CSS Animations |
| * Can only move from initial to final state — no intermediate steps * Can only run once * Require a trigger to run (like mouse hover) * Run forwards when triggered and in reverse when trigger is removed * Easier to use with JavaScript * Best for creating a simple change from one state to another | * Can move from initial to final state, with intermediate steps in between * Can run infinitely thanks to animation-iteration-count property * Can be triggered but can also run automatically * Can run forwards, in reverse, or alternate directions * More difficult to use with JavaScript * Best for creating a complex series of movements |

8)== and ===

|  |  |
| --- | --- |
| == | === |
| Checks if two values are equal | Checks if two values are equal |
| Doesn’t check if the data types are equal | Checks if the data type is also equal |
| Eg: let a=5;  let b=5;  let c='5';  let d=(a==b);  let e=(a==c);  console.log(d);  console.log(e);  Output:  True  True | Eg: let a=5;  let b=5;  let c='5';  let d=(a===b);  let e=(a===c);  console.log(d);  console.log(e);  Output:  True  False |

1. Sum of an array:

let arr1=[1,2,3,4,5];

let sum=0;

for(let i=0;i<=arr1.length-1;i++){

sum=sum+arr1[i];

}

console.log(sum);

Output:

15

1. Remove duplicate elements of an array

let arr2=[1,2,3,1,3,2,4,5,6];

let arr2new=[...new Set(arr2)];

console.log(arr2new);

Output:

[1,2,3,4,5,6]

1. INPUT TYPES IN HTML AND HTML5

## Following is a list of all types of <input> element of HTML.

|  |  |
| --- | --- |
| **type=" "** | **Description** |
| text | Defines a one-line text input field |
| password | Defines a one-line password input field |
| submit | Defines a submit button to submit the form to server |
| reset | Defines a reset button to reset all values in the form. |
| radio | Defines a radio button which allows select one option. |
| checkbox | Defines checkboxes which allow select multiple options form. |
| button | Defines a simple push button, which can be programmed to perform a task on an event. |
| file | Defines to select the file from device storage. |
| image | Defines a graphical submit button. |

****HTML5 added new types on <input> element. Following is the list of types of elements of HTML5****

|  |  |
| --- | --- |
| **type=" "** | **Description** |
| color | Defines an input field with a specific color. |
| date | Defines an input field for selection of date. |
| datetime-local | Defines an input field for entering a date without time zone. |
| email | Defines an input field for entering an email address. |
| month | Defines a control with month and year, without time zone. |
| number | Defines an input field to enter a number. |
| url | Defines a field for entering URL |
| week | Defines a field to enter the date with week-year, without time zone. |
| search | Defines a single line text field for entering a search string. |
| tel | Defines an input field for entering the telephone number. |

12)



OUTPUT:

